

# COL106: Data Structures and Algorithms

Ragesh Jaiswal, IIT Delhi

- How do we implement a Queue using Array?

# Data Structures

## Linked List

- Linked List: A collection of nodes with linear ordering defined on them.
  - Each node holds an element and points to the next node in the order.
  - The first node in the ordering is called the **head** and the last is called the **tail**.
  - The tail points to a **null** reference.
  - The data structure is accessed using a reference to the head node.
- Advantages of linked list:
  - The size of the data structure is roughly equal to the size of the elements that need to be stored. So, it is **space-efficient**.
  - The data structure is **resizable**.
  - “Shifting” not required as in the case of Arrays.



Figure : Visual representation of a Linked List

# Data Structures

## Linked List

- Linked List: A collection of nodes with linear ordering defined on them.
  - Each node holds an element and points to the next node in the order.
  - The first node in the ordering is called the **head** and the last is called the **tail**.
  - The tail points to a **null** reference.
  - The data structure is accessed using a reference to the head node.
- Give the mechanism for performing the following operations along with the running time:
  - Add an element at the beginning of the list:
  - Add an element at the end of the list:
  - Delete a particular node (given its reference):
  - Delete the first node containing element  $e$ :
  - Search element  $e$  in the linked list:
  - Remove the first element of the list:

# Data Structures

## Linked List

### Code

```
class Node{
    public int v;
    public Node next;
}
class LinkedList{
    public int size;
    public Node head;
    public LinkedList(){ size = 0; head = null; }
    public void AddAtHead(int e){...}
    public void AddAtTail(int e){...}
    public void DeleteNode(Node N){...}
    public void DeleteFirstNodeWithElement(int e){...}
    public void SearchElement(int e){...}
    public void DeleteFirst(){...}
}
```

# Data Structures

## Linked List

- Linked List: A collection of nodes with linear ordering defined on them.
  - Each node holds an element and points to the next node in the order.
  - The first node in the ordering is called the **head** and the last is called the **tail**.
  - The tail points to a **null** reference.
  - The data structure is accessed using a reference to the head node.
- Give the mechanism for performing the following operations along with the running time:
  - Add an element at the beginning of the list:  $O(1)$
  - Add an element at the end of the list:  $O(n)$
  - Delete a particular node (given its reference):  $O(n)$
  - Delete the first node containing element  $e$ :  $O(n)$
  - Search element  $e$  in the linked list:  $O(n)$
  - Remove the first element of the list:  $O(1)$

# Data Structures

## Linked List

- Linked List: A collection of nodes with linear ordering defined on them.
  - Each node holds an element and points to the next node in the order.
  - The first node in the ordering is called the **head** and the last is called the **tail**.
  - The tail points to a **null** reference.
  - The data structure is accessed using a reference to the head node.
- Give the mechanism for performing the following operations along with the running time:
  - Add an element at the beginning of the list:  $O(1)$
  - Add an element at the end of the list:  $O(n)$
  - Delete a particular node (given its reference):  $O(n)$
  - Delete the first node containing element  $e$ :  $O(n)$
  - Search element  $e$  in the linked list:  $O(n)$
  - Remove the first element of the list:  $O(1)$
  - **Reverse the list:**

End